

**5    Verfahren zum Schreiben von Speichersektoren in einem  
blockweise löschen Speicher**

Die Erfindung beschreibt ein Verfahren zum Schreiben von Speichersektoren in einzeln löschen Speicherblöcken, die eine Vielzahl von Speichersektoren enthalten, wobei auf die realen Sektoren jeweils mittels einer Zuordnertabelle zur Adressumsetzung von einer logischen Adresse in eine reale Blockadresse und eine reale Sektoradresse zugegriffen wird, und wobei jeweils, wenn ein Sektorschreibbefehl auszuführen wäre, der einen bereits beschriebenen Sektor betrifft, durch eine geänderte Adressumsetzung in einen Ausweichspeicherblock geschrieben wird.

Die weit verbreiteten nichtflüchtigen Halbleiterspeicher (Flashspeicher) sind in Blöcken und diese wiederum in Sektoren organisiert, wobei ein Block z.B. aus 256 Sektoren mit je 512 Bytes besteht. Die Speicher besitzen die Eigenschaft, dass neue Informationen nur sektorweise in vorher gelöschte Sektoren geschrieben werden können. Das Löschen erfolgt jeweils für einen Block für alle Sektoren gemeinsam. Das Schreiben eines Sektors in den Speicher dauert länger als das Lesen und eine Löschoperation eines Blockes erfordert eine lange Zeit, z.B. mehrere Millisekunden.

In der Patentanmeldung DE 102 27 256.5 ist ein Verfahren beschrieben, wie zum Schreiben eines Sektors ein Ausweichblock gesucht wird und die neue Information in dessen Sektoren geschrieben wird. Dabei wird in einer Bitliste als Sektormaske vermerkt, welche Sektoren neu in den jeweiligen Ausweichblock geschrieben wurden. Falls nun ein Sektor ein zweites Mal beschrieben werden muss, wird

- 2 -

ein neuer Ausweichblock gesucht und in diesen geschrieben. Dieses Verfahren hat den Nachteil, dass auch nach nur wenigen Schreibvorgängen in einen Ausweichblock ein neuer Ausweichblock gesucht und Sektoren umkopiert werden müssen.

- 5    Weiterhin besitzen neuere Speicherchips die Eigenschaft, dass die Sektoren innerhalb eines Blocks nur sequentiell aufsteigend beschrieben werden können. Wenn also nicht nacheinanderliegende Sektoren beschrieben werden sollen, muss auch ein neuer Ausweichblock gesucht und schon beschriebene Sektoren kopiert werden.
- 10   Die Kopiervorgänge stellen relativ langsame Schreibvorgänge dar, welche die Geschwindigkeit des Speichersystems beeinträchtigen. Aus dem US-Patent 5,835,935 ist bekannt, bei Schreibanforderungen jeweils den nächsten freien Sektor in einem Block zu nutzen. Damit werden die Blöcke zwar sequentiell beschrieben, aber es wird jeweils
- 15   die Zuordnungstabelle zwischen logischer und physikalischer Sektoradresse angepasst, was jeweils einen weiteren Schreibvorgang erfordert.

- Es ist Aufgabe der Erfindung, ein Verfahren zu offenbaren, das die beschriebenen Nachteile vermeidet und die Geschwindigkeit des Speichersystems verbessert.

- Gelöst wird diese Aufgabe dadurch, dass die Schreibvorgänge von Sektoren in den Ausweichspeicherblock sequentiell erfolgen und die Position des jeweiligen Sektors im Ausweichblock in einer Sektorabelle gespeichert wird.

- Vorteilhafte Ausführungsformen sind in den Unteransprüchen angegeben.

- 30   Beim Schreiben eines Sektors wird dieser durch eine logische Adresse adressiert. Diese Adresse kann in eine logische SektorAdresse und eine logische BlockAdresse aufgespaltet werden. Das Speichersystem ist in

reale Speicherblöcke, die zusammen gelöscht werden können, und in reale Sektoren, die einzeln beschrieben werden können, organisiert. In einer Zuordnertabelle werden die logischen Blockadressen realen Speicherblockadressen zugeordnet. Die logischen Sektoradressen 5 werden ohne eine Adressumsetzung direkt als reale Sektoradressen genutzt. Einige Speicherblöcke sind als Ausweichblöcke reserviert, die bei den Schreiboperationen genutzt werden können. Für den jeweils zu beschreibenden Speicherblock wird ein Ausweichblock zugeordnet und im internen Merkspeicher des zum Speichersystem 10 gehörenden Speichercontrollers werden die Adressen der Ausweichblöcke und die zugehörigen Listen geführt, in welche Sektoren geschrieben wurde. Nun ist es vorteilhaft, die geänderten Sektoren eines Blockes sequentiell in den zugehörigen Ausweichblock zu schreiben und die Liste als Sektortabelle zu führen die angibt, in 15 welchen Sektor des Ausweichblocks der betreffende Speichersektor geschrieben wurde.

Dabei ist die Sektortabelle in einer Ausführung als Indextabelle aufgebaut, auf die mit der realen Sektoradresse (die gleich der 20 logischen Sektoradresse ist) zugegriffen wird. Der jeweilige Tabelleneintrag an dieser Position gibt denjenigen Sektor im Ausweichblock an, in den der entsprechende Speichersektor geschrieben wurde.

25 Die Indextabelle wird zu Anfang mit den höchstmöglichen Werten (FF) initialisiert. Ein solcher Wert gibt an, dass der zugehörige Sektor noch unverändert im originalen Speicherblock steht.

In einer anderen Ausführung der Sektortabelle ist diese als Suchtabelle 30 aufgebaut, in der in jedem Tabelleneintrag die reale Sektoradresse mit der jeweiligen gültigen Sektorposition im Ausweichblock angegeben ist. Dabei ist es vorteilhaft, die Suchtabelle nach realen

- 4 -

Sektoradressen zu ordnen, um schnell die aktuelle Position eines Speichersektors zu finden.

In beiden Ausführungen der Sektortabelle ist es günstig, die Position des Sektors im Ausweichblock auch im Verwaltungsbereich des Sektors mit zu speichern. Dadurch kann bei einem Wiederanlauf des Speichersystems die Sektortabelle aus den im Verwaltungsbereich gespeicherten Sektorpositionen im Merkspeicher des Speichercontrollers erneut aufgebaut werden. Bei einem solchen Wiederanlauf wird dann die höchste Positionsnummer des zugehörigen Sektors in die Sektortabelle übernommen.

In der Ausführung des Verfahrens ist eine typische Sektoranzahl für einen Speicherblock 256 Sektoren. In der Ausführung mit einer Indextabelle wäre dann diese 256 Byte lang. In der Ausführung mit einer Suchtabelle wäre diese dann 32 Byte lang.

Bei einer Ausführung der Tabelle mit 32 Bytes kann die Sektormaske für 256 Sektoren mit einer Suchtabelle für 16 Sektoren überlagert werden.

Durch verschiedene Schreiboperationen in einen Ausweichblock wird die zugehörige Sektortabelle gefüllt. Sobald das Ende der Sektortabelle erreicht wird, wird ein neuer Ausweichblock gesucht und in diesen die gültigen Sektoren aus dem originären Speicherblock und dem bisherigen Ausweichblock zusammenkopiert. Danach nimmt dieser Block die Stelle des originären Speicherblocks an, indem der Eintrag in der Zuordnertabelle entsprechend aktualisiert wird. Die bisherigen Speicher- und Ausweichblöcke werden zum Löschen freigegeben und dienen nach dem Löschen als neue Ausweichblöcke.

Um für den jeweiligen Speicherblock die günstigste Schreibverwaltung einzustellen, wird in der Zuordnertabelle zu jeder logischen Blockadresse ein Strategiemerker mitgeführt. Dieser Merker gibt an,

- 5 -

- ob bei Schreiboperationen zuletzt mit einer bitweisen Sektormaske oder einer byteweisen Sektortabelle gearbeitet wurde. Diese Verwaltungsart des Ausweichblocks wird auch für die nächsten Schreiboperationen beibehalten. Die Strategiemerker werden 5 günstigerweise zunächst mit der Angabe „Sektormaske“ initialisiert, da die meisten Schreiboperationen rein sequentiell ablaufen und dann die Verwaltung über eine Sektormaske am schnellsten erfolgt.
- Bei Speichersystemen vom Typ „FAT-Filesystem“ ist es aber 10 vorteilhaft, die Speicherblöcke, welche die FAT enthalten, sogleich mit dem Strategiemerker „Sektortabelle“ zu initialisieren, da es bei einem solchen Speichersystem zu häufigen Schreiboperationen in die gleichen Speichersektoren kommt. Damit wird die Geschwindigkeit des Speichersystems von Beginn an gesteigert.
- 15 Wird während des Betriebs des Speichersystems festgestellt, dass einzelne Speichersektoren öfter überschrieben werden, ist es günstig, die Verwaltung des Ausweichblocks von einer Sektormaske auf eine Sektortabelle umzustellen. Dabei wird dann auch der Strategiemerker 20 in der Zuordnertabelle entsprechend nachgeführt.
- Gibt es keine Duplikate von Sektoren, ist es günstig, auf die Sektormaske zur Verwaltung des Speicherblockes umzustellen.
- Die Ausgestaltung der Erfindung ist in den Figuren beispielhaft 25 beschrieben.  
Fig. 1 zeigt die Zuordnung der Speicherblöcke zu den logischen Adressen  
Fig. 2 zeigt die Schreibverwaltung mittels einer Indextabelle  
Fig. 3 zeigt die Schreibverwaltung mittels einer Suchtabelle  
30 Fig. 4 zeigt die Zusammenfassung eines Speicherblocks mit einem Ausweichblock zu einem neuen Speicherblock

- 6 -

In der Fig. 1 ist die logische Adresse LA dargestellt, die sich aus der logischen Blockadresse LBA und der logischen Sektoradresse LSA zusammensetzt. Mit der logischen Blockadresse LBA wird auf die Zuordnertabelle ZT zugegriffen, in der die zugehörige reale Adresse  
5 RA gespeichert ist. Die reale Adresse RA setzt sich aus der realen Blockadresse RBA und der realen Sektoradresse RSA zusammen. Dabei wird die reale Sektoradresse RSA nicht gespeichert, da die logische Sektoradresse LSA auch als reale Sektoradresse RSA genutzt wird. Weiterhin ist in der Zuordnertabelle ZT der jeweils zu der  
10 logischen Blockadresse gehörige Strategiemerker SF gespeichert, der angibt, ob mit einer Sektormaske oder einer Sektortabelle gearbeitet wird. Die zu einer logischen Adresse LA gehörige reale Adresse RA zeigt auf einen Speicherblock SB, der für die Speicheroperation mit der logischen Adresse LA genutzt wird.

15

In der Fig. 2 ist ein Speicherblock SB und ein zugehöriger Ausweichblock AB gezeigt, bei dem die Speicheroperationen über eine Indextabelle IT gesteuert werden. Auf die Indextabelle IT wird  
20 über die reale Sektoradresse RSA zugegriffen. Der Tabelleneintrag gibt entweder die tatsächliche Position des zugehörigen Sektors in dem Ausweichblock AB an oder zeigt an, dass der entsprechende Sektor im Speicherblock SB noch gültig (G) ist. Dazu ist in der Indextabelle IT an diesen Positionen ein höchster Wert (FF)  
25 eingetragen. Nicht mehr gültige Sektoren, bei denen sich ein neuerer Sektorinhalt im Ausweichblock befindet, sind im Speicherblock SB mit (X) dargestellt. Im Ausweichblock AB wird sequentiell in die Sektoren geschrieben, was durch den Pfeil angedeutet ist. Es können sich mehrere Sektorinhalte zu einer logischen Sektoradresse LSA im  
30 Ausweichblock befinden. In diesem Beispiel betrifft es den Sektor 0, der an den Positionen 0 und 2 im Ausweichblock AB eingetragen ist. Die zugehörige Position 0 in der Indextabelle IT bezeichnet den gültigen Sektor. Um beim Systemlauf die Tabelle richtig

- 7 -

rekonstruieren zu können, wird die aufsteigende Reihenfolge der Sektoren eingehalten. Wenn beim Lesen des Ausweichblocks ein Sektor, in diesem Beispiel Sektor 0, mehrfach gefunden wird, ist der Sektor mit der höchsten Position im Block der gültige Sektor.

5

In Fig. 3 ist die gleiche Situation wie in Fig. 2 mit einer Suchtabelle ST dargestellt. Sie ist nach logischen Sektoradressen LSA geordnet, hier die Sektoren 0, 2, 5. In jedem Tabelleneintrag befindet sich zugeordnet der gültige Sektor im Ausweichblock AB. Nicht in der 10 Suchtabelle ST verzeichnete Sektoren befinden sich weiterhin in dem Speicherblock SB und sind hier mit G gekennzeichnet.

In Fig. 4 ist die Zusammenführung eines Speicherblocks SB mit einem Ausweichblock AB zu einem neuen Speicherblock NB 15 dargestellt. Die veränderten Sektoren, die im Ausweichblock AB verzeichnet sind, werden in die entsprechenden Sektoren im neuen Speicherblock NB kopiert. Die restlichen Sektoren werden aus dem originären Speicherblock SB in den neuen Speicherblock NB kopiert. Dieser enthält damit nur gültige Sektoren, hier mit G gekennzeichnet.

**Bezugszeichen**

AB	Ausweichblock
FF	höchster Wert
G	Gültiger Sektor
IT	Indextabelle
LA	Logische Adresse
LBA	Logische Blockadresse
LSA	Logische Sektoradresse
NB	Neuer Speicherblock
RBA	Reale Blockadresse
RSA	Reale Sektoradresse
SB	Speicherblock
SF	Strategiemerker
ST	Suchtabelle
X	ungültiger Sektor
ZT	Zuordnertabelle

## Patentansprüche:

1. Verfahren zum Schreiben von Speichersektoren in einzeln löschen**5**baren Speicherblöcken (SB), die eine Vielzahl von Speicher-sektoren enthalten, wobei auf die realen Sektoren jeweils mittels einer Zuordnertabelle (ZT) zur Adressumsetzung von einer logischen Adresse (LA) in eine reale Blockadresse (RBA) und eine reale Sektoradresse (RSA) zugegriffen wird, und wobei jeweils, wenn ein Sektorschreibbefehl auszuführen wäre, der einen bereits beschriebenen Sektor betrifft, durch eine geänderte Adressum-**10**setzung in einen Ausweichspeicherblock (AB) geschrieben wird, dadurch gekennzeichnet, dass die Schreibvorgänge von Sektoren in den Ausweichspeicherblock (AB) sequentiell erfolgen und die Position des jeweiligen Sektors im Ausweichblock in einer Sektortabelle gespeichert wird.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die geänderte Adressumsetzung über einen Datensatz mit realer Blockadresse (RBA) und Sektortabelle im internen Merkspeicher eines Speichercontrollers erfolgt.**20**
3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Sektortabelle als Indextabelle (IT) aufgebaut ist, wobei die reale Sektoradresse (RSA) als Index dient und an der jeweiligen Tabellenposition die gültige Sektorposition im Ausweichblock (AB) angegeben ist.**25**
4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass ein höchstmöglicher Wert in der Indextabelle (IT) zu einer Sektoradresse (RSA) angibt, dass der zugehörige Sektor unverändert im ursprünglichen Speicherblock (SB) steht.**30**
5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Sektortabelle als Suchtabelle (ST) aufgebaut ist, wobei in jedem

- 10 -

Tabelleneintrag die reale Sektoradresse (RSA) mit der jeweiligen gültigen Sektorposition im Ausweichblock (AB) angegeben ist.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass die

5 Suchtabelle (ST) nach realen Sektoradressen (RSA) geordnet ist.

7. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die

Position des Sektors im Ausweichblock (AB) auch im  
Verwaltungsbereich des Sektors mit gespeichert wird.

10

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass die  
Sektortabelle eines Blocks beim Wiederanlauf des Speichersystems  
aus den im Verwaltungsbereich gespeicherten Sektorpositionen  
erneut aufgebaut wird.

15

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, dass beim  
Wiederanlauf die Position des Sektors mit der höchsten  
Positionsnummer in die Sektortabelle übernommen wird.

20

10. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass ein  
Speicherblock 256 Sektoren enthält und die zugehörige  
Indextabelle (IT) 256 Byte lang ist.

25

11. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass ein  
Speicherblock 256 Sektoren enthält und die zugehörige  
Suchtabelle (ST) 32 Byte lang ist.

30

12. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass, sobald  
die Sektortabelle gefüllt ist, ein neuer Ausweichblock gesucht  
wird und in diesen die gültigen Sektoren aus dem originären  
Speicherblock und aus dem bisherigen Ausweichblock  
zusammenkopiert werden.

- 11 -

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass der neue Ausweichblock in der Zuordnertabelle als originärer Speicherblock eingetragen wird und die bisherigen Speicher- und Ausweichblöcke zum Löschen freigegeben werden.

5

14. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass in der Zuordnertabelle zu jeder logischen Blockadresse ein Strategiemerker mitgeführt wird, der angibt, ob für diese eine Sektormaske oder eine Sektorabelle zuletzt genutzt wurde.

10

15. Verfahren nach Anspruch 14, dadurch gekennzeichnet, dass der Strategiemerker mit der Angabe „Sektormaske“ initialisiert wird.

15

16. Verfahren nach Anspruch 15, dadurch gekennzeichnet, dass bei einer Formatierung des Speichersystems als FAT-Filesystem die Speicherblöcke für die FAT mit der Angabe „Sektorabelle“ initialisiert werden.

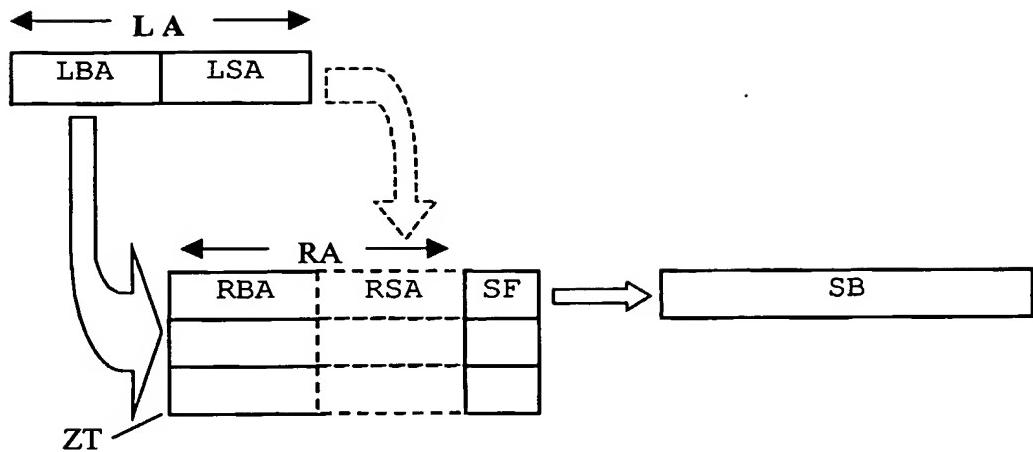
20

17. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass falls nur wenige Sektoren im Ausweichblock geschrieben wurden und einer dieser Sektoren nochmals geschrieben werden soll, die Verwaltung des Ausweichblocks von der Sektormaske auf die Sektorabelle umgestellt wird.

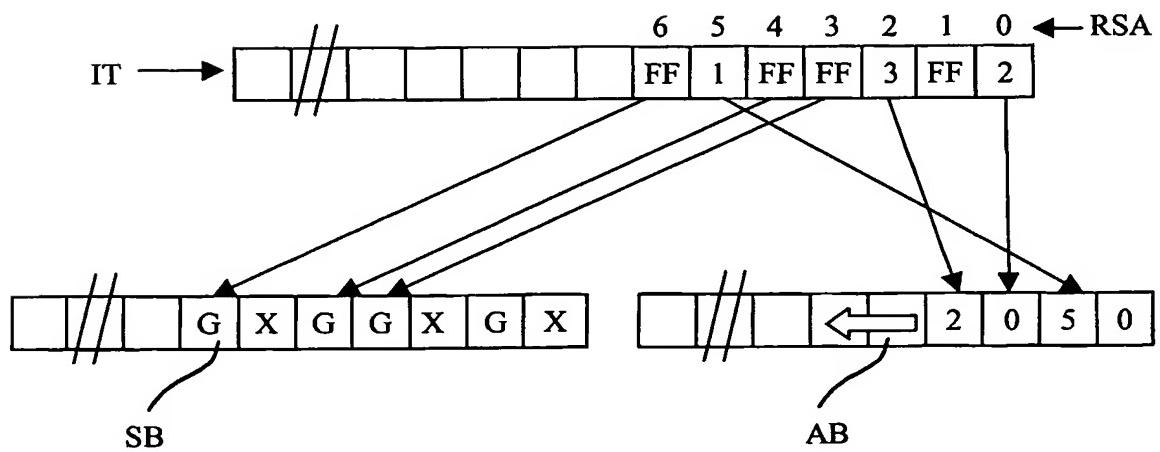
25

- 1 / 2 -

**Fig. 1**

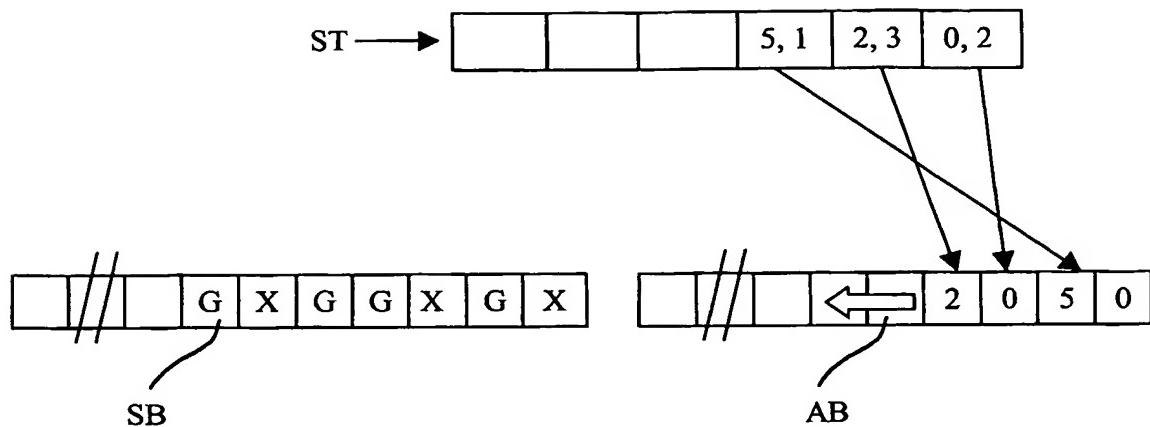


**Fig. 2**



- 2 / 2 -

**Fig. 3**



**Fig. 4**

